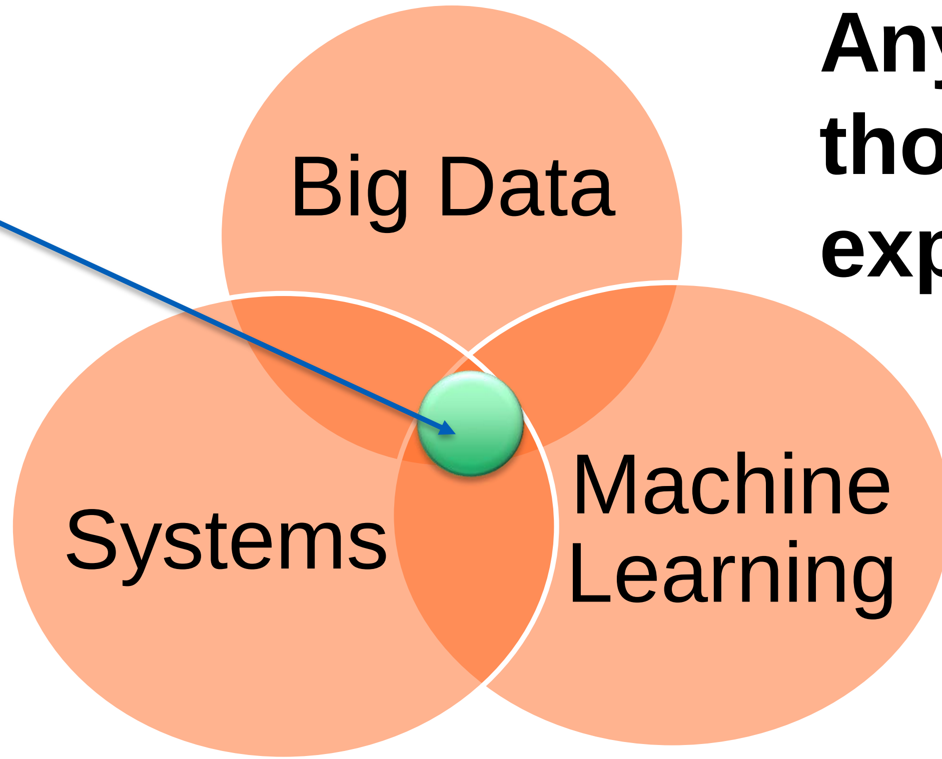
Aalto University
School of Science

Robustness, reliability, resilience and elasticity for Big Data/Machine Learning systems

Hong-Linh Truong
Department of Computer Science
linh.truong@aalto.fi, https://rdsea.github.io

Our focus in this course

The focus



Any idea, thought, expectation?

Content

- **Big Data/ML background**
- **Design for robustness, reliability, resilience and elasticity**
- **An elasticity-based approach for R3E**

Recap

Big data and Machine learning systems

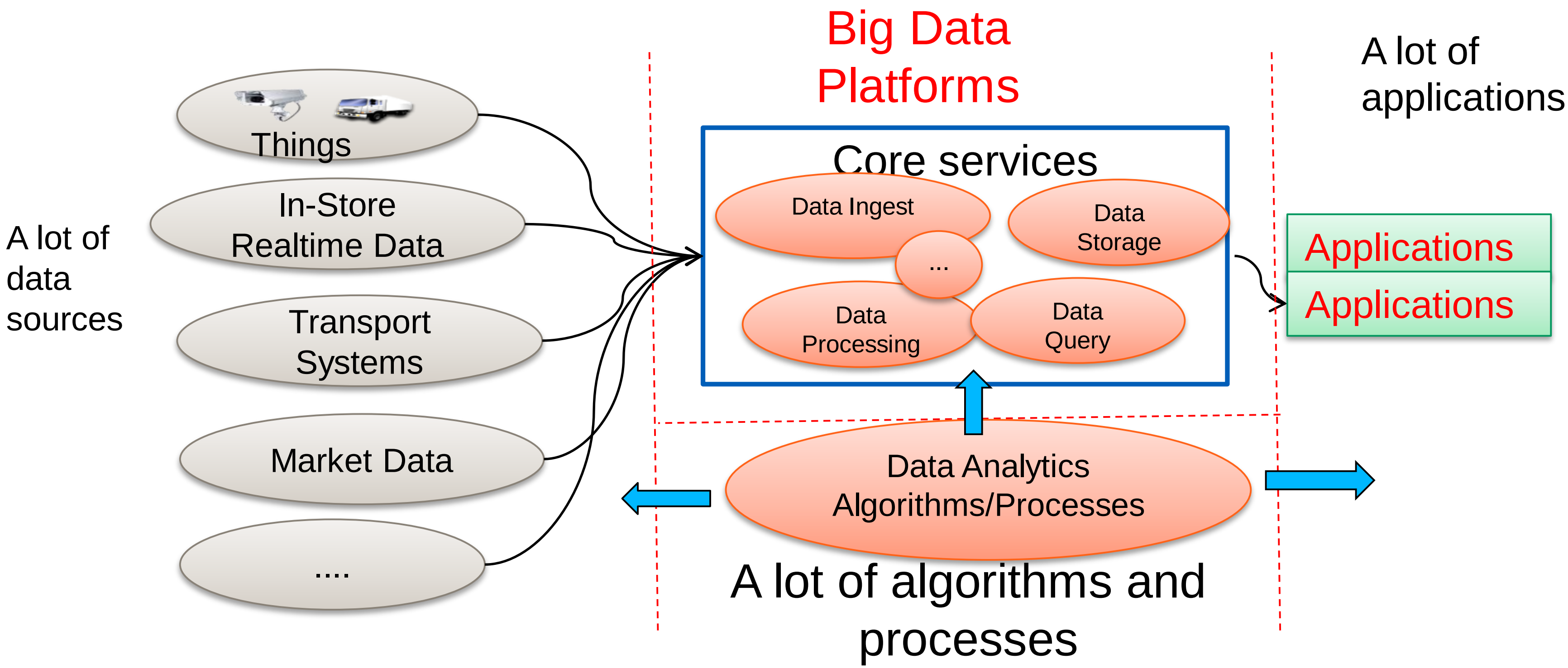
System view: common characteristics of big data and ML systems?

- **(Static) system structures and functions**
  - Include components, algorithms, relationships, possible input/output data
  - As a whole, sub-systems, and individual parts
- **Computing and data infrastructures/platforms?**
  - Virtual machines, containers, brokers, storage, data
- **Runtime quality/capability**
  - Fault-tolerance, high-performance, high availability, secure, etc.

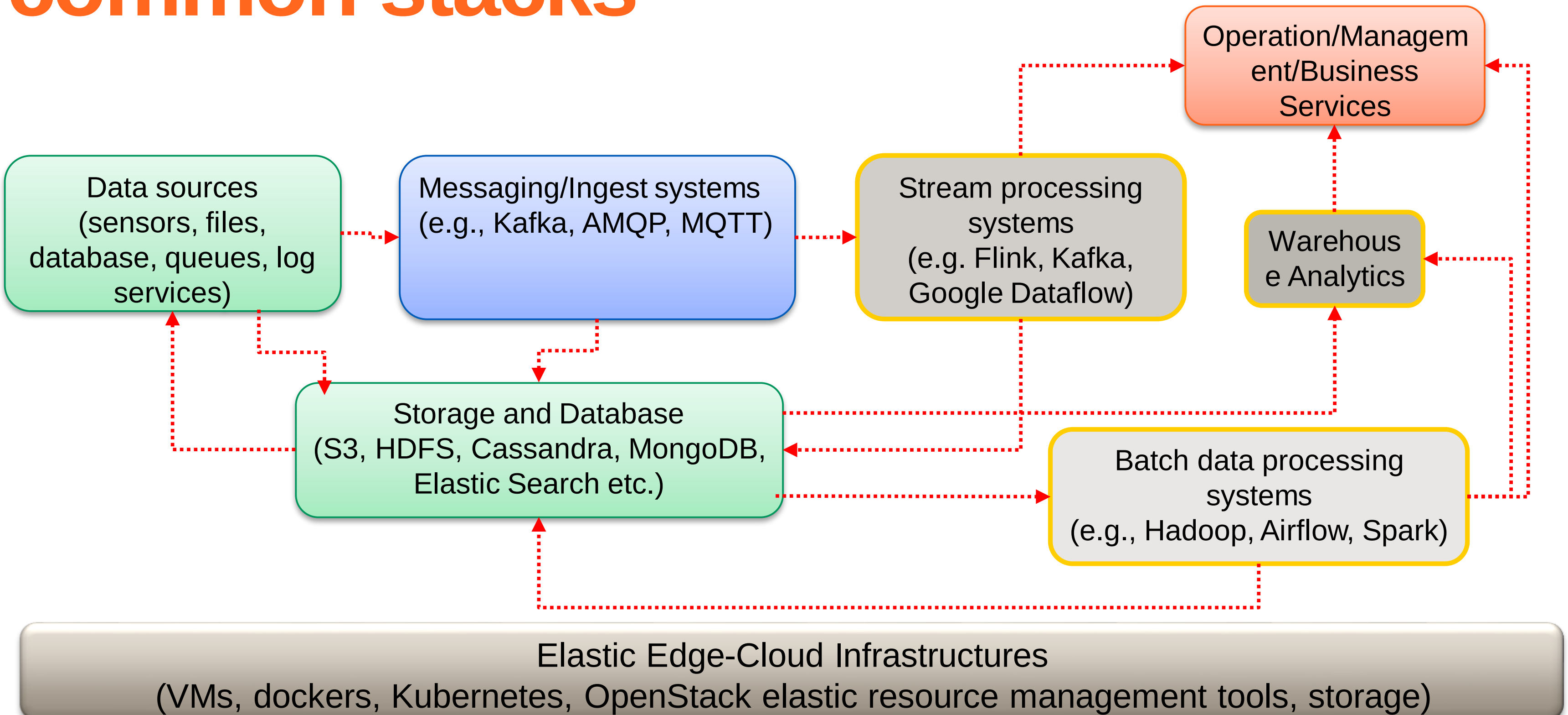
Big data with V*

- **Volume:**
  - big size, large data set, massive of small data
- **Variety:**
  - complexity of different formats and types of data
- **Velocity:**
  - generating speed, data movement speed
- **Veracity:**
  - quality is very different (timeliness, accuracy, etc.)

A bird view of big data platforms



Big data at large-scale: example of common stacks

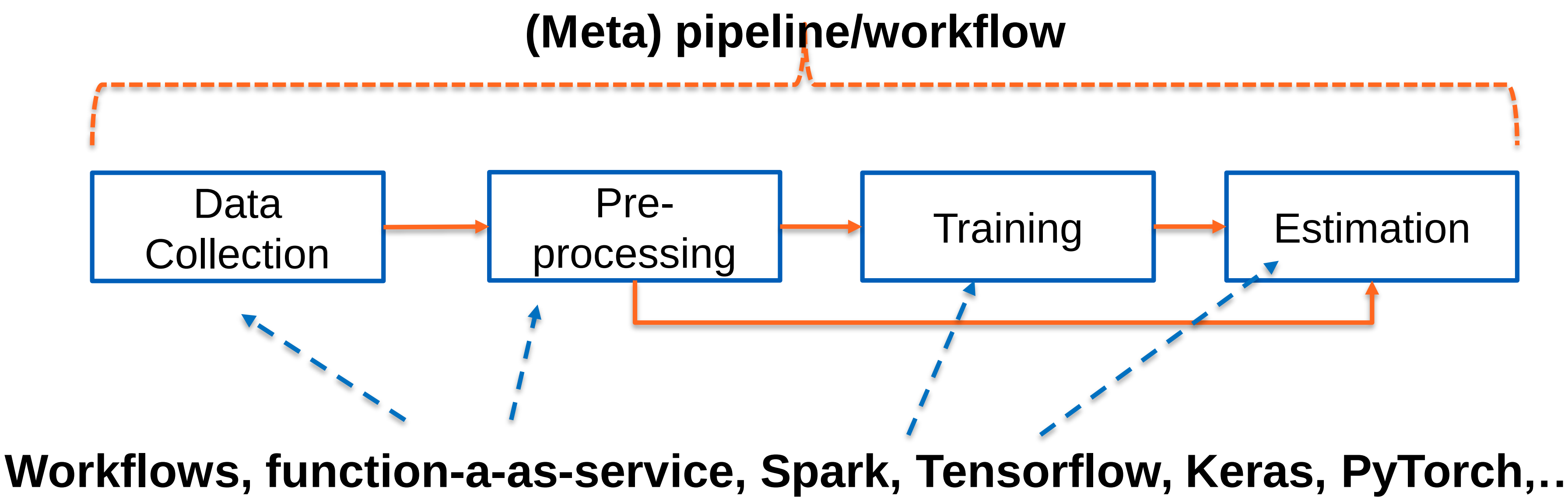


ML systems

- **Components in machine learning**
  - machine learning algorithms can be considered “data processing”
  - there are many other components for data-preparation, data management, experiment management
- **ML pipelines**
  - complex structured components, (meta)workflows
- **Data**
  - models, training data, data to be learned!
  - experiment settings and experiment data
  - from the big data platforms viewpoint: they are all data!

ML workflows

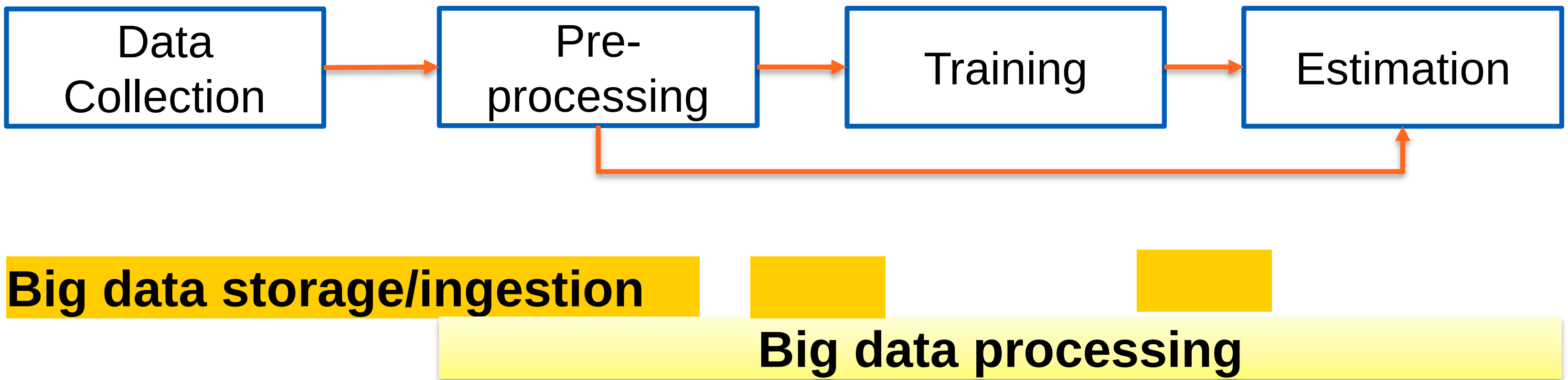
- **Two possible levels:**
  - meta-workflow or pipeline
  - inside each phase: pipeline/workflow or other types of programs



Structures: examples of common components

- **Data collection, ingestion, verification**
- **Algorithms and service service/components**
  - Serving platforms and infrastructures
- **Configuration and process execution management**
- **Monitoring and analysis**
- **Resource management**

Examples of common components in big data and ML systems

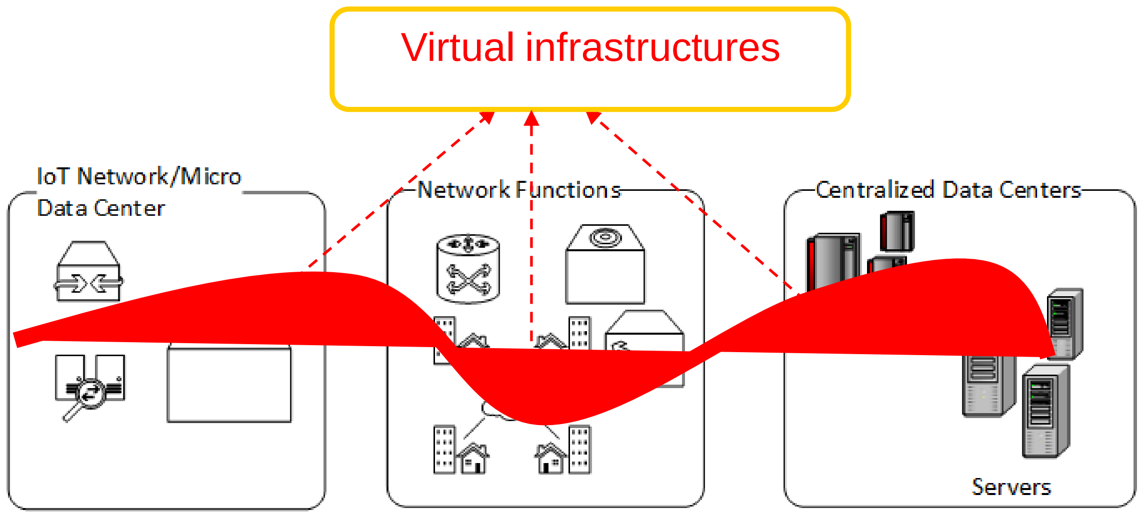


Resource management, workflow execution, data management tools, etc.

Computing and data infrastructures

View: end-to-end resource slice

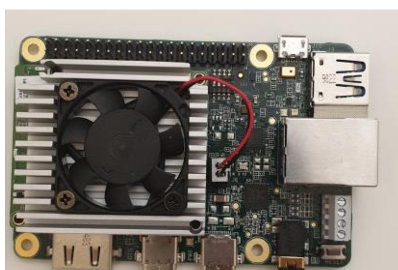
End-to-end Resource slice for big data/ML



Edge-Cloud systems

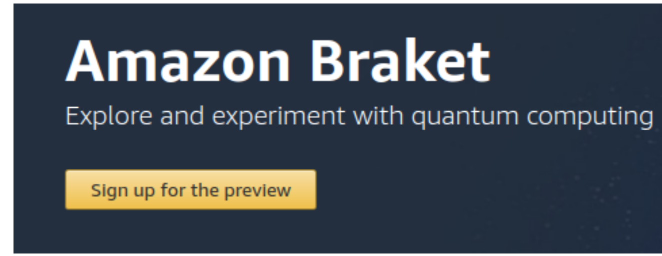
New types of edge and edge-cloud

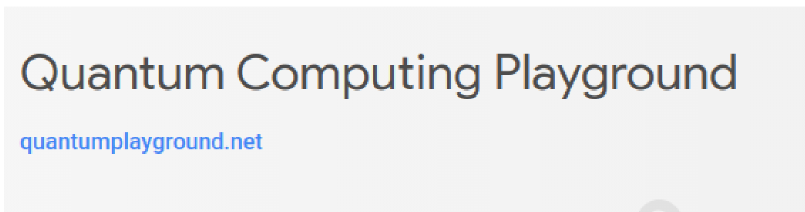
Coral with Edge TPU System-on-Module, Google Edge TPU ML accelerator coprocessor

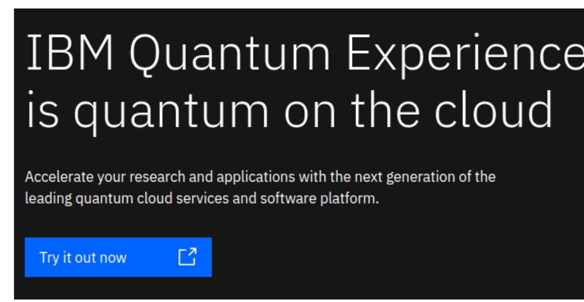


Jetson NVIDIA (GPU+CPU)

New (hype?) quantum computing services for ML







Assumption and setting for this course

- **You know how to build big data/ML systems**
  - it DOES NOT mean to be master in both big data AND ML
- **You focus on YOUR "systems/applications"**
- **We try to help to look from systems viewpoint**
  - which are key abilities that we should define, design, monitor, and measure?
  - how to enable flexibility and execution management?
  - how to prepare for "future"/"emerging" infrastructures?
  - which are tools and frameworks that help our engineering?

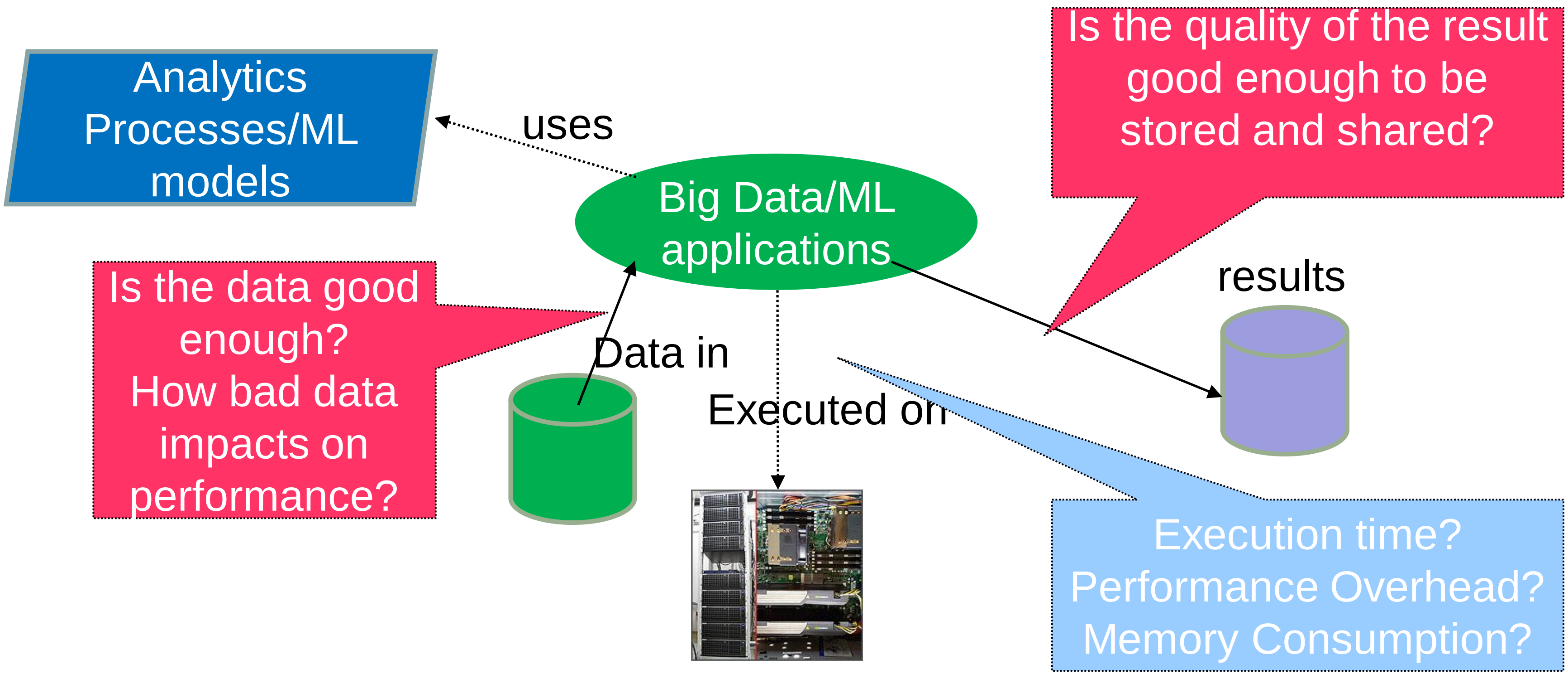
Issues in our concerns

- **Development**
  - testing, experimenting, benchmark, optimization, cost management
- **Resources**
  - execution atop multiple computing frameworks suitable for ML, such as Clouds, Supercomputing, edge, …
- **(Runtime) Ability/Quality Assurance**
  - specification, monitoring and assurance of performance, availability, costs, reliability, etc.

Runtime abilities/capabilities

Can you name some runtime abilities/capabilities that are important for your big data/ML systems?

Quality of Analytics (QoA)



QoA = {quality of result, performance, cost}

Our focus – R3E

- **Robustness**
  - ability to cope with errors
- **Reliability**
  - ability to function according to the indented specification (in a proper way)
- **Resilience**
  - "ability to provide the required capability in the face of adversity"(https://www.sebokwiki.org/wiki/System_Resilience)
- **Elasticity**
  - ability to stretch and return to normal forms (under external forces)

Robustness

- **In ML**
  - overfitting/underfitting
  - transfer learning
  - machine learning in an open-world
    - *how to deal with OOD (out-of-distribution) situations?*
  - when we can decide to stop training if performance/robustness does not improve?
- **Big data**
  - hwow to deal with erroneous and bad data?

Reliability

- **System reliability versus "reliable service" (quality of analytics)**
- **System reliability**
  - reliable infrastructures, components, networks, ...
- **"Reliable service" → reliable analysis**
  - without failure, with specified performance
- **Some hard problems**
  - have good and enough data, clean data
  - robust pipelines without degraded performance and accuracy

Resilience

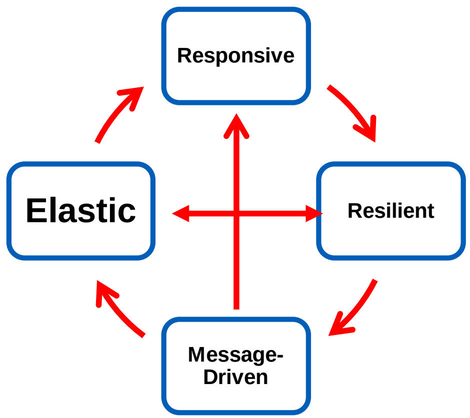
- **Common issues in resilience**
  - distributed software and systems bugs
  - system attacks
- **Some specific issues in big data/ML systems**
  - bias in data
  - well-known problems in adversary attacks in ML phases

Elasticity

- **Add and remove resources**
  - CPUs, memory, data, networks, ...
- **Dynamic changes of algorithms**
- **Shift computation between edge and cloud infrastructures dynamically**
  - cloud data centers, edge systems and edge-cloud systems
- **Remove data to improve performance**
- **Hyperparameter tuning tradeoffs**

Reactive systems – an architectural style for R3E?

Reactive systems



Source: https://www.reactivemanifesto.org/

For enabling R* abilities:

- **Responsive: quality of services**
- **Resilient: deal within failures**
- **Elastic: deal with different workload and quality of analytics**
- **Message-driven: allow loosely coupling, isolation, asynchronous**

Do we need to treat them all equally in all your design?

Multi-dimensional view for optimization

- **Structures**
  - multiple algorithms, components, and services can be combined in different ways
- **Resources**
  - data, networks, machines, humans
  - cross-infrastructures/providers
- **Runtime quality/capabilities**
  - customized based on context and requirements

Our goal in the course is to seek for generic techniques and solutions (not specific optimizations for a specific applications)

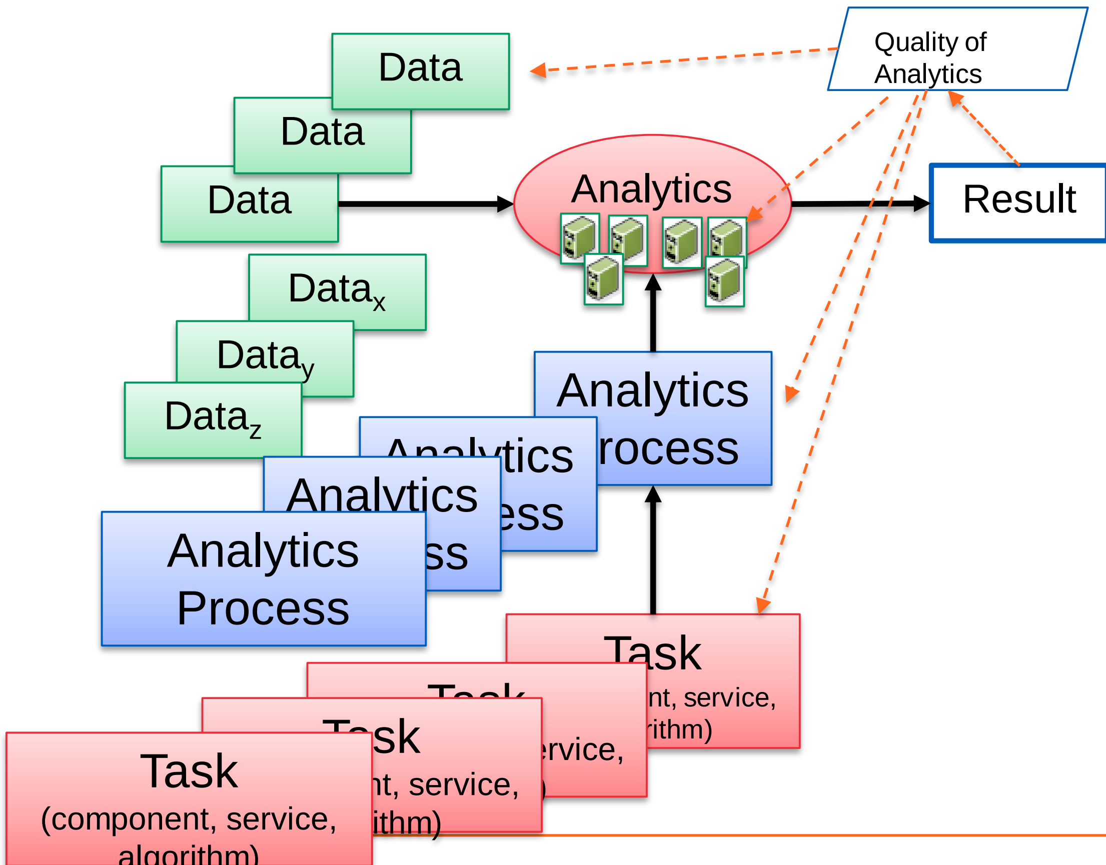
An Approach with Elasticity Principles for R3E

Elasticity

- **Demand elasticity**
  - elastic demands from consumers
- **Output elasticity**
  - multiple outputs with different price and quality
- **Input elasticity**
  - elastic data inputs, e.g., deal with opportunistic data
- **Elastic pricing and quality models associated resources**

But the key thing is to be "elastic" in the way we should optimize our big data/ML systems

Elasticity in (big) data analytics



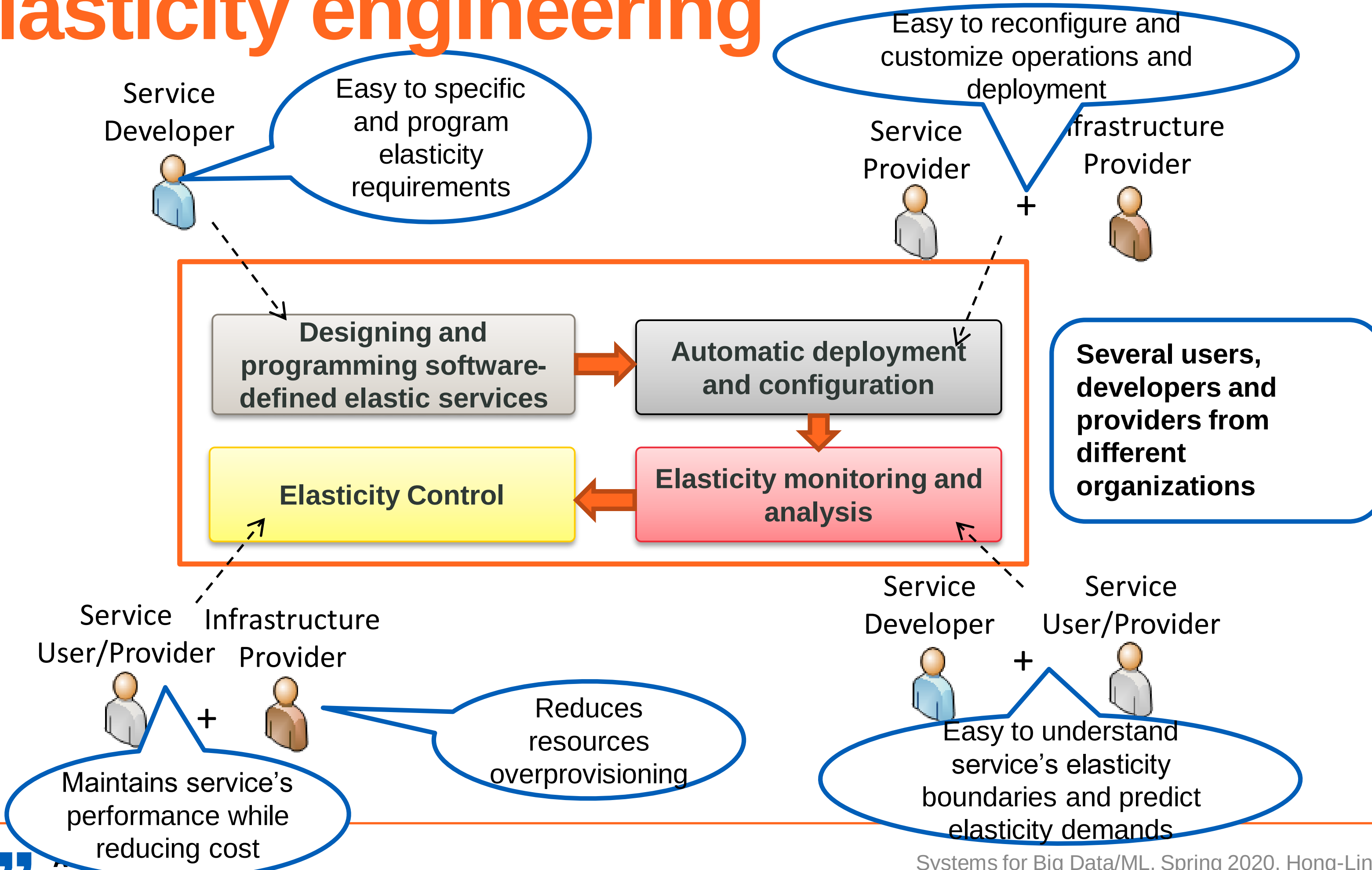
- **More data → more compute resources (e.g. more VMs)**
- **More types of data → more, different tasks → more analytics processes**
- **Change quality of analytics**
  - Change quality of data
  - Change response time
  - Change cost
  - Change types of result (form of the data output, e.g. tree, table, story)

Multi-dimensiona' Elasticity

Example

You can build your own dimensions

Elasticity engineering



Fundamental building blocks for the elasticity

- **Conceptualizing and modeling elastic objects (and their instances) and execution environments**
  - Diverse types of artifacts and their runtime in a similar manner
- **Defining and capturing elasticity primitive operations associated with elastic objects and environments**

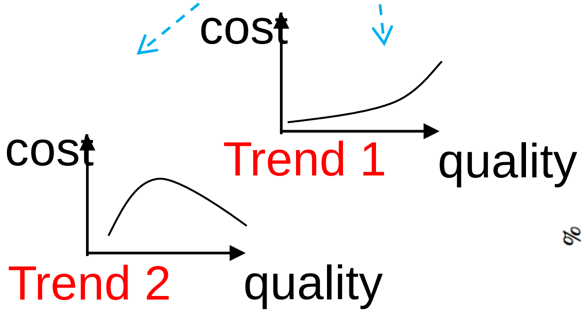
- **Recommending and Programming elastic objects**
  - An elastic system can be built from elastic objects
- **Runtime deploying, control, and monitoring techniques for elastic objects**

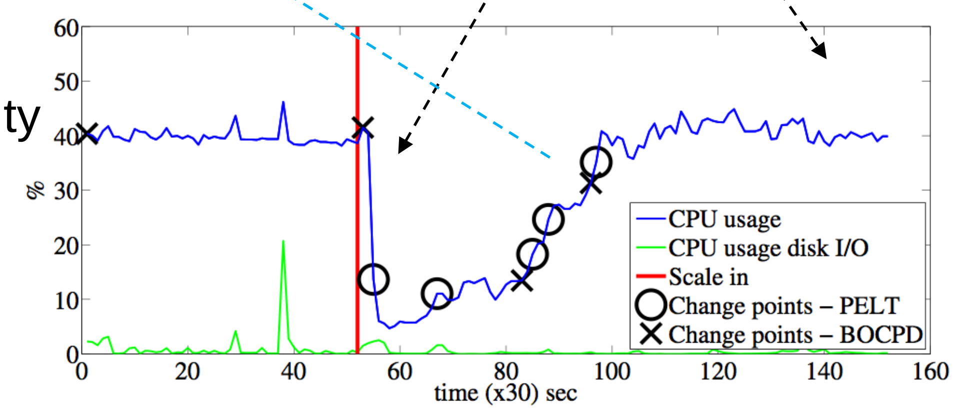
Detecting elasticity

Elasticity Space Func (from when to when?)

Elasticity Trend/Prediction Func

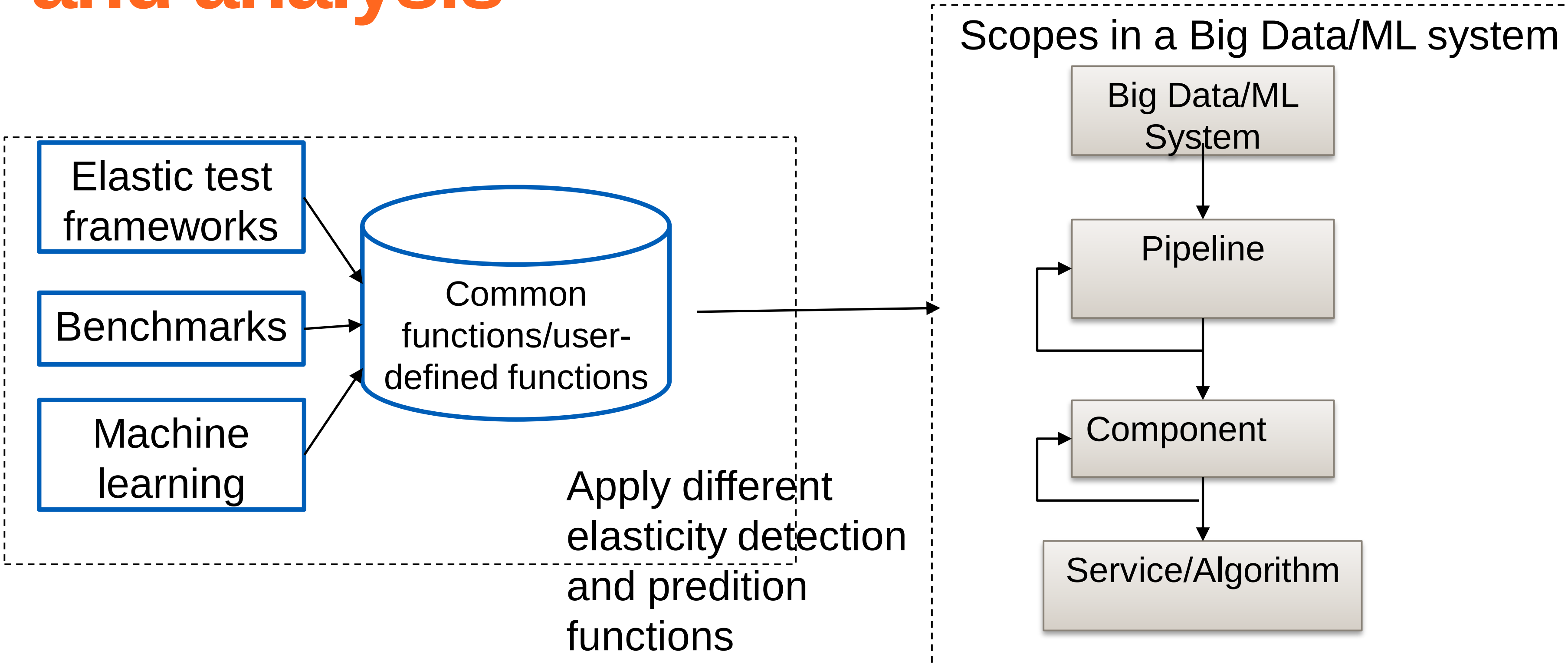
Change point detection algs



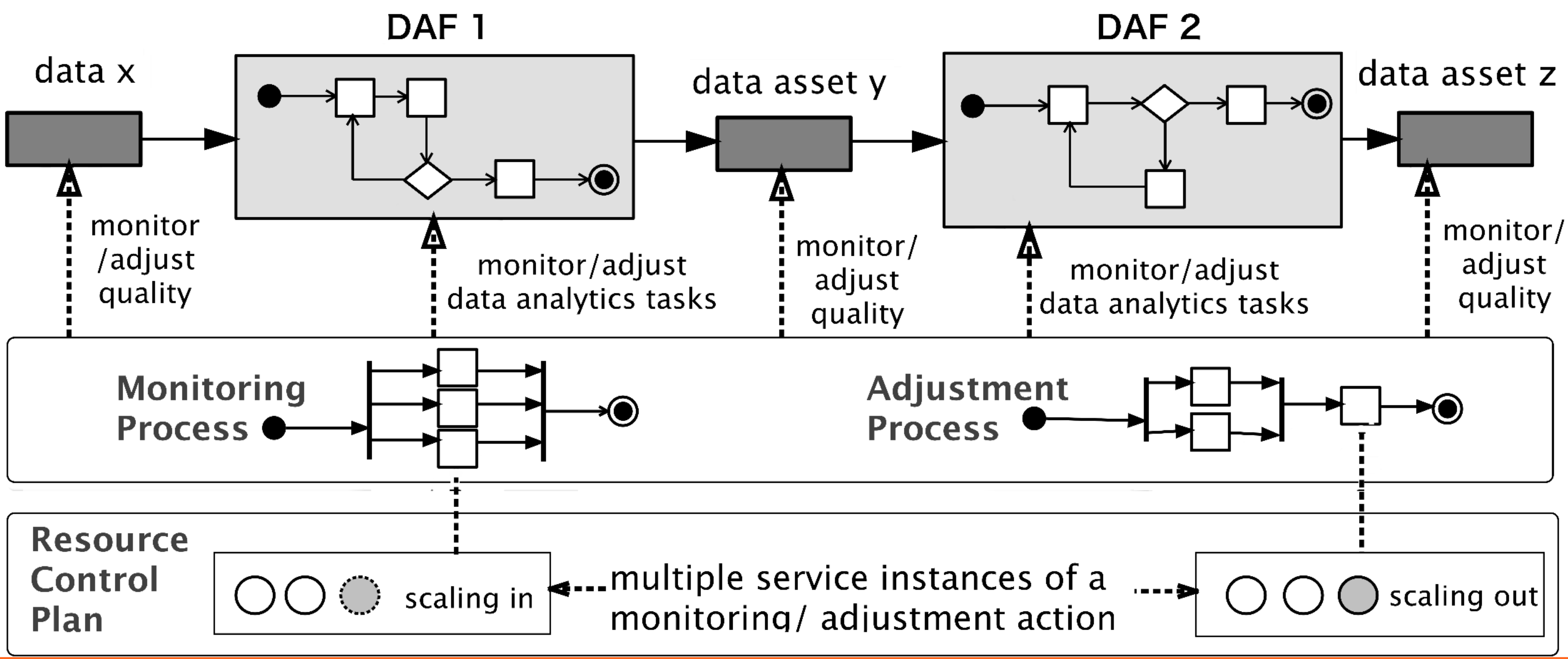


Alessio Gambi, Daniel Moldovan, Georgiana Copil, Hong Linh Truong, Schahram Dustdar: On estimating actuation delays in elastic computing systems. SEAMS 2013: 33-42

Multi-level cross platforms monitoring and analysis



Using Elasticity Management Process to ensure QoA



Examples: Optimizing QoA for Retail

Quality of analytics management of data pipelines for retail forecasting

Title:	Quality of analytics management of data pipelines for retail forecasting
Author(s):	Kreics, Krists
Date:	2019-08-19
Language:	en
Pages:	54+3
Major/Subject:	Data science
Supervising professor(s):	Truong, Hong-Linh
Thesis advisor(s):	Ervasti, Mikko; Luukkonen, Teppo
Keywords:	machine learning, offline learning, data pipelines, quality of analytics, apache airflow
Location:	Archive
	OEV Publication only in digital format

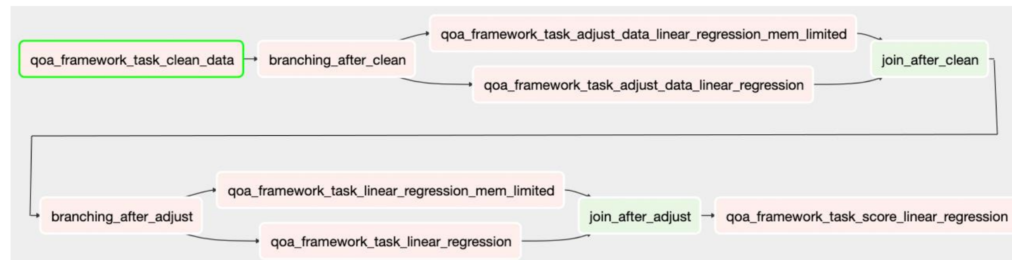
https://aaltodoc.aalto.fi/handle/123456789/39908

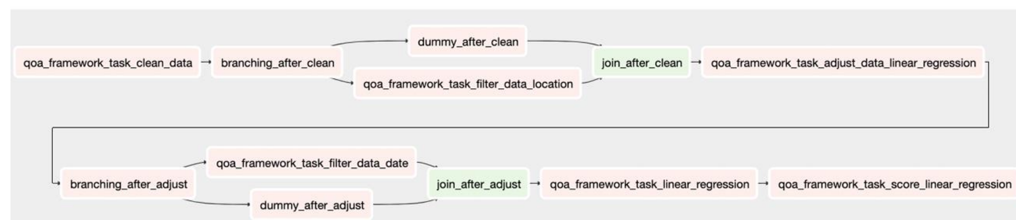
Training industrial retail forecast ML

Forecast where to put marketing information, example of data

date	id	name	volume	price	cost	promo	category_net	margin	category1	category2	location	sales
07/01/2018	100	Chicken	38144.0	3.79	2.7	0	451692.0	0.25	Meat	Food	Helsinki	144565.76
14/01/2018	100	Chicken	36420.0	3.79	2.66	0	414342.0	0.25	Meat	Food	Helsinki	138031.8
21/01/2018	100	Chicken	35322.0	3.79	2.66	0	381854.0	0.25	Meat	Food	Helsinki	133870.38

- **Metrics:**
  - Data size, R square value, time, and cost
- **Pipelines**
  - Tune pipelines with QoA primitive actions





Initial results

- **Running with Airflows in Amazon EC2**
- **Apply different actions to change “store” (domain objects) and computing resources**
- **Real improvement (from the domain expert) with 1 million rows case**

  13.3% lower accuracy and 44% shorter time, R squared value was 9.5% lower → could good enough results for 50% of total store locations

The application-aware data reduction strategy and cost-accuracy tradeoffs may be more intelligently made based on knowledge of the application domain.

Study log for this week

Think about

- **What does it mean R3E for** *YOUR big data and machine learning systems***?**

Then

- **in your experience/work, which ones of R3E concern you most? Why? What would you do? What do you look for?**
- **1 page – submit into the mycourses for comments/feedback (keep it in your git)**

Thanks!

Hong-Linh Truong
Department of Computer Science

rdsea.github.io